AD-A037 127 BROWN UNIV PROVIDENCE R I DIV OF APPLIED MATHEMATICS F/G 5/7
ABDUCTION MACHINES AND LANGUAGE ACQUISITION.(U)
APR 76 S SHRIER
N00014-75-C-0461
UNCLASSIFIED NL

| OF |
AD
A037127

END
DATE
FILMED
4-77

(15) Contract N00014-75-C-0461
Office of Naval Research and Brown University

(12)

(11) April 1976

(12) 32p.

(6) Abduction Machines and Language Acquisition

(9) Progress Report,

by

(10) S. Shrier
Brown Univ.

This paper deals with a model of language syntax acquisition. It is assumed that the artificial language has a finite description which we hope to discover on the basis of a finite sample of sentences. Specifically excluded is the simple formulation of the observations actually made, although a naïve description is highly desirable. In the terminology of learning models, an insightful model is to be preferred over the rote learning exemplified in a list.

Proposal for the study of this problem was posed in the paper "Pattern Conception" by Miller and Chomsky [1957]; this paper elaborated on the virtues of finite state automata models. An early description of a machine to carry out grammar discovery is given by Solomonoff in [1957]. The various guises and disguises of this problem are found in artificial intelligence, human cognitive studies, pattern recognition, linguistics and in systems theory under labels such as inductive inference, automaton identification and grammatical inference. The fine exposition by Fu [1974] in a chapter entitled "Grammatical Inference for Syntactic Pattern Recognition" surveys many approaches and also contains 49 references. Additional references can be found in Trakhtenbrot and Barzdin' [1973].

As far as modelling of acquisition is concerned, little attention has been focussed on possible natural solutions. The following investigation is motivated toward the presentation of model models. The investigation follows the paths initiated by Grenander [1974] in his abductory induction models. If we restrict consideration to regular grammars, the number of possible grammars is overwhelmingly large for even small size alphabets and modest numbers of variables. This implies that models for

analogy to real world phenomena which exhibit language acquisition ability cannot be based on enumerative inference or finite search techniques. for this reason, we also seek an alternative to direct implementations of maximum likelihood solutions which select one grammar over another on probabilistic criteria determined by the solution of large scale linear systems.

The model consists of two components; the first component is a teacher in a dual role. The teacher acts as a generative grammar which produces strings in a syntax-controlled probability language (Grenander [1967]). The random structure of the language is induced by imposing probabilities on the rules of the grammar. In this way, the production of some strings can be inhibited while the production of others can be made more likely. This adds to the teacher's grammar a facet of linguistic performance although it does not increase the generative power of the grammar. The teacher also serves as an acceptor of strings; it can judge whether or not a sentence presented to it by the learner is within its competence.

The second component, called the learner, consists of two principal procedures which carry out the construction of a copy of the teacher. The first procedure or phase is devoted to the classification of words into equivalence classes on the basis of grammatical substitutability; this procedure is in principle infinitary. These classes are familiar in structural linguistics where they are called families (Kulagina [1958]) or categories (Miller and Chomsky [1963]) although we do not use these classes in quite the same way. They are introduced here to reduce the combinatorial complexity of the constructions.

The first phase begins with the assumption that all words are in one equivalence class; this is the tabula rasa with which the learner begins. The discovery of the classes is carried out by the resolution of the dictionary into the classes which form the required partition. The teacher randomly generates strings which are presented to the learner as a training sequence. For each string in the training sequence, a word is selected according to a weighted probabilistic strategy; this might be thought of as an attention function. This string is used to either strengthen the learner's belief about the selected word's membership in its present grammatical class or it is used to introduce a new hypothesis to be entertained about the relation of this word to the sought for partition of the dictionary. The term abduction, introduced by C. S. S. Peirce to describe the starting of a hypothesis, is applied to describe this process which either changes the class membership of the selected word or forms a new class with this selected word. All classes formed are characterized by a fixed representative word called a prototype.

The second phase carries out the discovery of the syntactic variables and the rewrite rules which govern these variables. Initially in this phase, the learner has a tabula rasa with respect to variables; the discovery of variables proceeds in a analogous manner to the phase one process. Each string in the training sequence is analyzed to some preset depth to determine initial string equivalence classes, i. e. the syntactic variables. Each initial string can be decoded with respect to the word class partition determined in phase one as described above. This indexing scheme is utilized to implent efficiently the process which determines the

partition of initial strings into the sought for syntactic variables. A subsequent encoding of initial strings is a representation of the rewrite rules. A simple tally scheme computes the experimental frequency defined probabilities.

The model described above is a blueprint for a language discovery machine. Such a machine has been implemented in *AED*\360; this machine has been tested on a fragment English grammar and on several formal grammars. The fragment English grammar consists of 87 rules on 52 words in 23 classes; the rules govern the 18 syntactic variables in this syntax-controlled probability grammar. The expected sentence length is 7.05 words. In a typical experiment, 115 sentences were "listened to" by the learner to determine 20 of the 23 classes and to correctly classify 48 of the words in the dictionary. After 27 sentences were analyzed to a depth of 8 words, 17 of the 18 variables were discovered; when the depth was increased to 15, 20 additional sentences were generated before the remaining variable was discovered. The graphs in figures (1) and (2) illustrate the learning characteristics exemplified by the word class discovery procedure.

Figure (1)

NUMBER OF DISCOVERED WORD CLASSES v. NUMBER OF SENTENCES

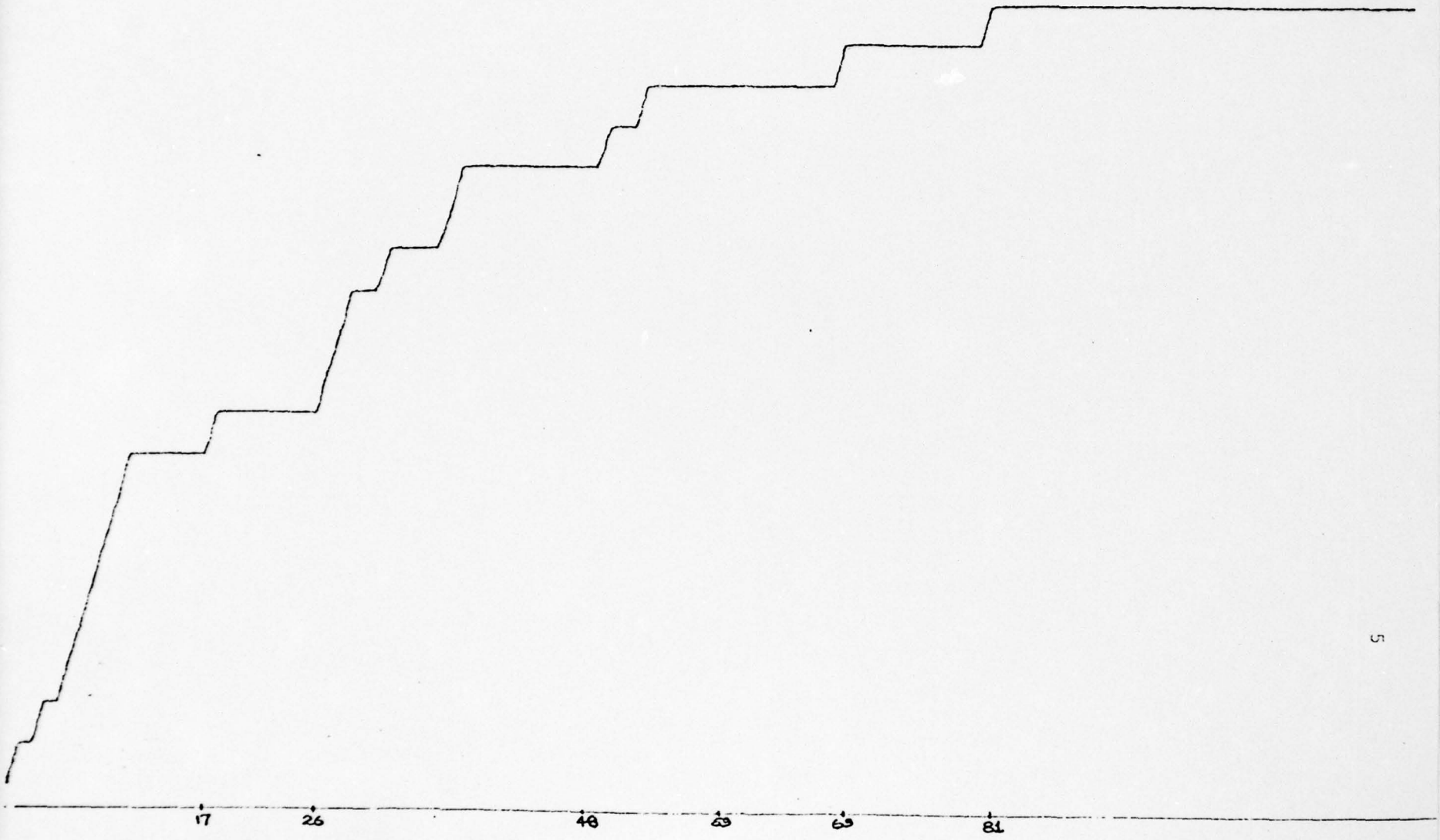
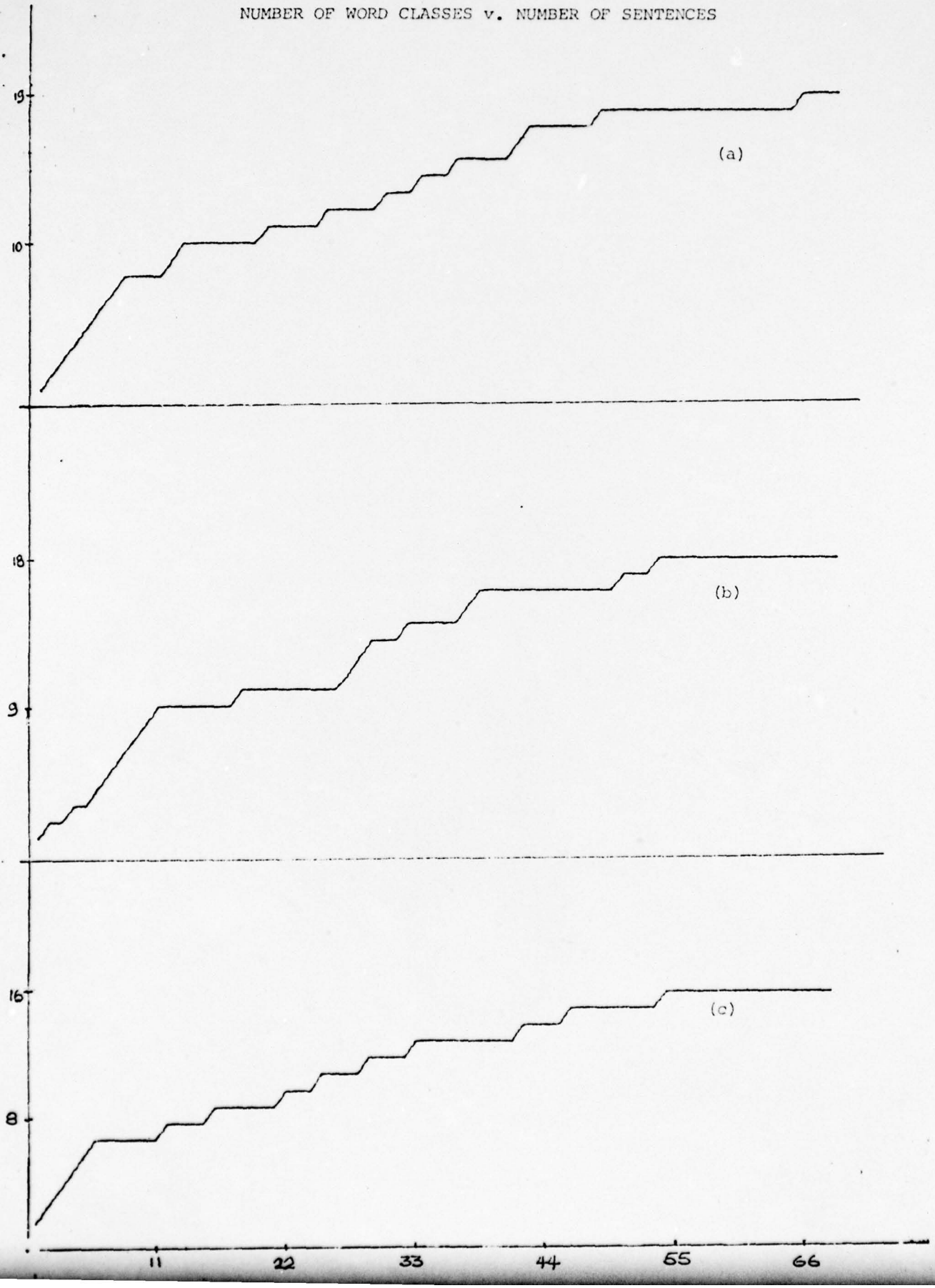


Figure (2)


NUMBER OF WORD CLASSES v. NUMBER OF SENTENCES
19
10
(a)
18
9
(b)
16
8
(c)
11
22
33
44
55
66

Bibliography

Fu, K. S. [1974], Syntactic Methods in Pattern Recognition, Academic Press, New York.

Grenander, U. [1967], "Syntax-controlled probabilities", R. P. A. 2.

Grenander, U. [1974], "Abduction machines that learn syntactic patterns", R. P. A. 25.

Kulagina, O. S. [1958], "Ob odnom sposobe opredelenija grammaticeskih ponjatii na baze teorii mnozestv", Problemy Kibernetiki t. 1, 203-214.

Miller, G. A. and N. Chomsky [1957], "Pattern conception", ASTIA AD 110076, Cambridge.

Miller, G. A. and N. Chomsky [1963], "Finitary models of language users", Handbook of Mathematical Psychology v. 2, Wiley, New York.

Solomonoff, R. J. [1957], "An inductive inference machine", IRE National Convention Record V.

Trakhtenbrot, B. A. and Ja. M. Barzdin' [1973], Finite Automata: Behavioral Synthesis, American Elsevier, New York.

May 1976

Abduction Machines and Language Acquisition

Progress Report

by

S. Shrier

The Word Class Discovery Algorithm

Experimental Results and Data Analysis

1. After t sentences have been generated, we have observations (r,z_r), for r=1,2,3,. . .,t, where z_r might represent the total number of words correctly classified (W) or the total number of discovered word classes (WC) after the rth sentence has been processed. These data are the result of a probabilistic process so that each experiment from "BIRTH" will have different characteristics; it is hoped that these differences might be slight. It has been suggested that Gosset's students' t tests might be applied to the results of several experiments. Another procedure might be to average over the several experiments the datum observed at each r and then to apply the data analysis described below to this smoothed, preprocessed data. Neither of these has yet been done nor explored further at this time.

2. The graphical representation of the data in figure 1, which is depicted as continuous for convenience, visually suggests learning curve properties. The data plotted on a

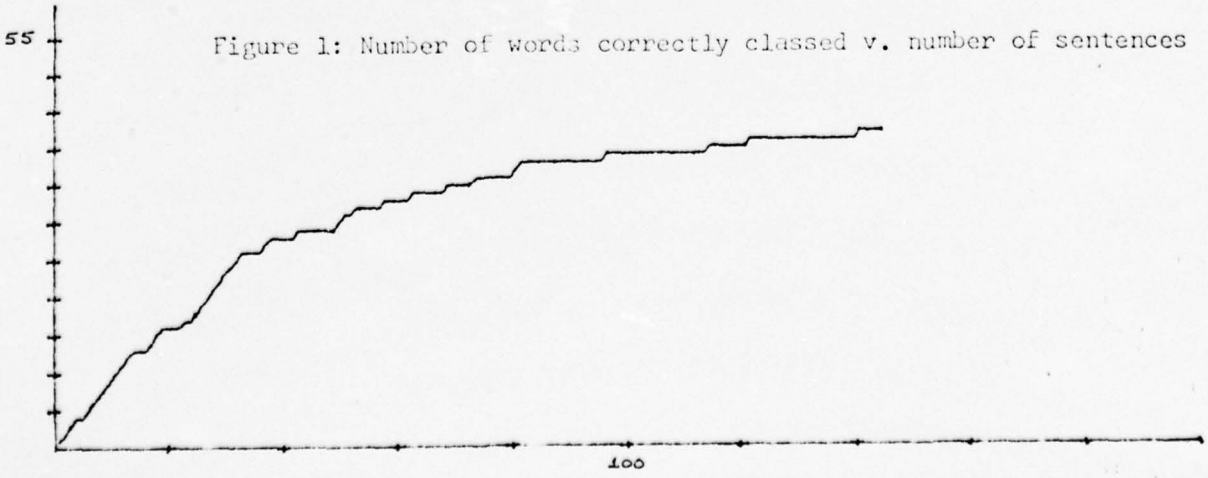
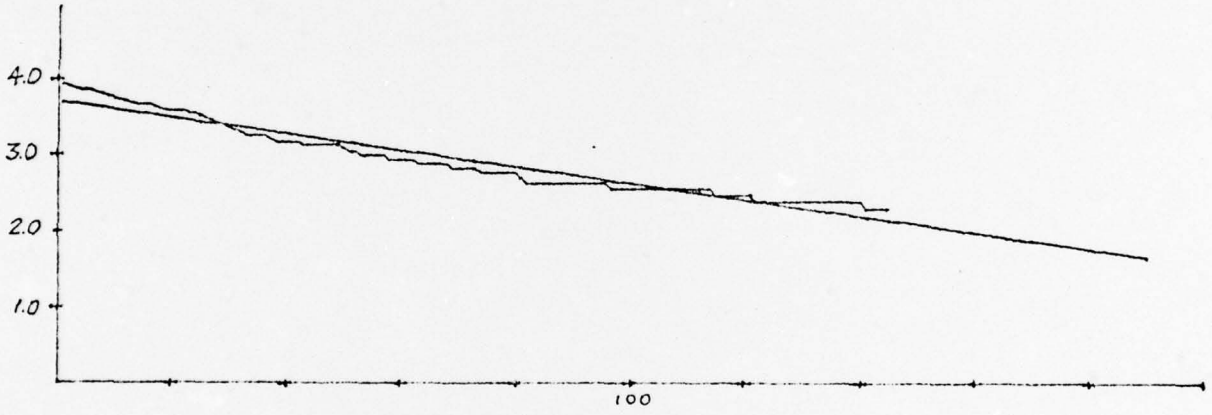
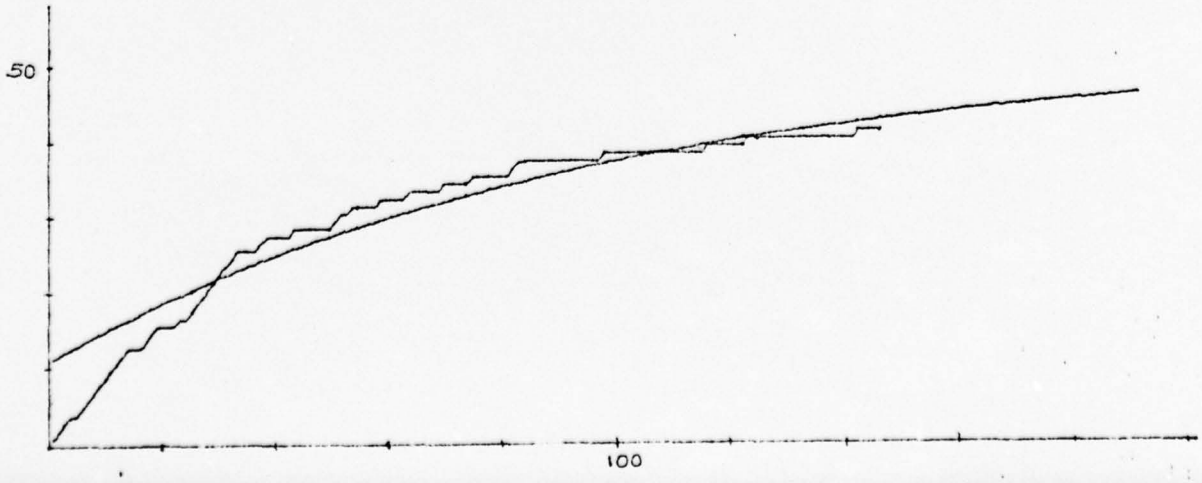


Figure 1: Number of words correctly classed v. number of sentences

semi-log scale appears in figure 2a. A least squares straight line fit to $(r, \ell n(z_a - z_r))$, where z_a denotes an asymptotic value, was determined and is superposed in figure 2a and was used to obtain the "fit" shown in figure 2b. The true asymptotic value, which in these experiments is of course known a priori, is increased by one so that in semilog coordinates (r,0) corresponds to "all words correctly classified".

Figures 2a & 2b





However, this fit does not have the vitally important property which underlies the theoretical model: the tabula rasa hypothesis which makes natural a constrained least squares fit to force the curve through the point (1,1); this point corresponds to the initial equivalence class.

3. That is, letting $y_r = \ell n(z_a - z_r)$, we seek m and b to minimize $M = \sum_{r=1}^{t}(y_r - mr - b)^2$, subject to the constraint $y_1 = \ell n(z_a - 1) = m + b$. This enables us to determine that

$$m = [\underline{y} - \underline{1}\ell n(z_a - 1)]\cdot\underline{u}/(\underline{u}\cdot\underline{u})$$

$$b = -m + \ell n(z_a - 1),$$

where the t-vectors $\underline{u} = (r - 1)$, $\underline{y} = (y_r)$ and $\underline{1} = (1,1,\ldots,1)$ have been introduced for clarity. The line is thus $y = mr + b$ which yields the experimental formula to be

$$z = z_a - \exp(mr + b) = z_a - (z_a - 1)\exp(-m)\exp(mr).$$

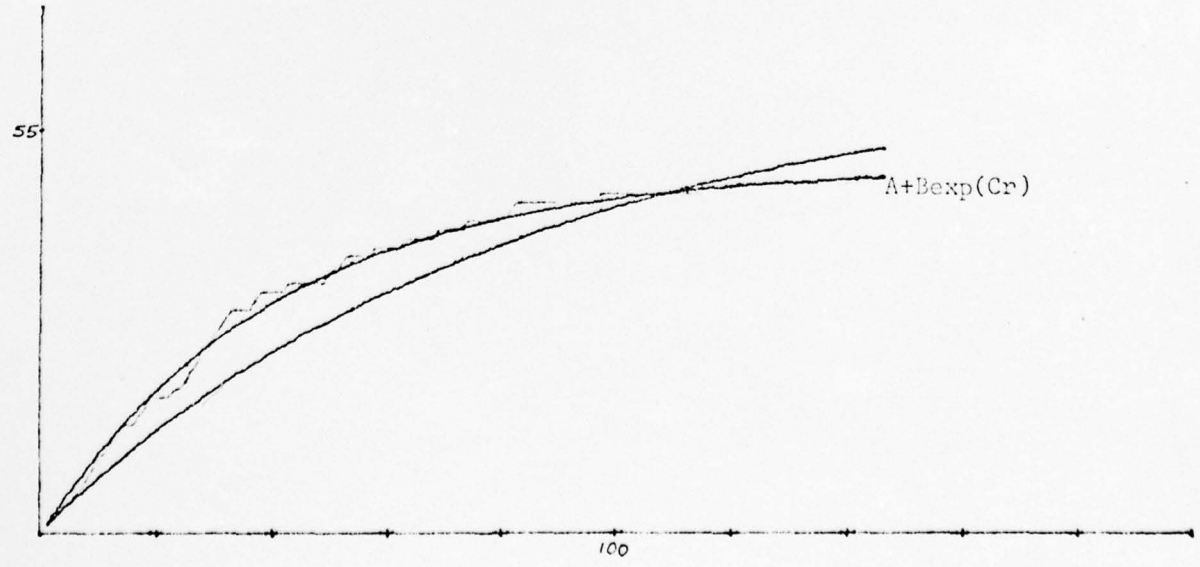
The data presented in table 1 has been analyzed according to this plan; the results of this analysis appear in table 1.

4. The Euclidean norm of the residual vector can of course be reduced by the direct application of the least squares procedure to the form $A + Bexp(Cr)$ with the observed data. The constrained problem: find A, B and C to minimize $M = \sum_{r=1}^{t}(z_r - A - Bexp(Cr))^2$ subject to the tabula rasa constraint $z_1 = A + Bexp(C)$ can be readily computed as follows: the transcendental equation in C

$$(\underline{z}-\underline{1})\cdot(\underline{v}' - [(\underline{v}\cdot\underline{v}')/(\underline{v}\cdot\underline{v})]\underline{v}) = 0$$

where $\underline{v} = (exp(C)-exp(Cr))$ and $v' = \frac{dv}{dc}$ is solved for C and this value is used to determine $B = -\underline{v}\cdot(\underline{z}-\underline{1})/(\underline{v}\cdot\underline{v})$ and $A=1 - Bexp(C)$. This has been done and the results are also recorded in table 1 and depicted graphically in figure 3.

Figure 3(original data shown in red)



freq	#correct	cum freq
1	1	1
1	2	2
1	3	3
2	4	5
1	5	6
1	6	7
1	7	8
1	8	9
1	9	10
1	10	11
1	11	12
1	12	13
3	13	16
1	14	17
1	15	18
4	16	22
2	17	24
1	18	25
1	19	26
1	20	27
1	21	28
1	22	29
1	23	30
1	24	31
1	25	32
4	26	36
1	27	37
5	28	42
7	29	49
1	30	50
2	31	52
5	32	57
5	33	62
6	34	68
5	35	73
7	36	80
1	37	81
15	38	96
18	39	114
7	40	121
19	41	140
5	42	145

Section 3: $m=-.013216$

$\exp(-.013216)=.98687$

$z = 52.\ [1 - .99382(.98687)^r]$

Section 4: $A=42.440$

$B=-42.493$

$C=-.025089$

$z = 42.44\ [1 - 1.00125(.97522)^r]$

TABLE 1

5. A conjecture based on visual observation is that the learning characteristics fall between the values determined in section 3 and section 4; the sought for characteristics would better represent both the early (rapid) learning and the asymptotic qualities viz., the time to determine all the words correctly. A heuristic procedure based on the observed data could be implemented; this would especially be of use if predictive qualities of the model were desired. In the least squares procedure, a weighting function appropriately chosen would improve the fit. We mention here that the true asymptotic value (e.g.the number of words to be classified), which is known a priori, denoted here by z_a could be introduced as an additional constraint to reformulate the least squares problem as: find C to minimize $\sum_{r=1}^{t}(z_r - z_a[1 - (1-1/z_a)\exp(C(r-1))])^2$. This leads to a transcendental equation in C

$$[(\underline{z} - \underline{1}z_a)/(1 - z_a) - \exp(C\underline{u})]\cdot\underline{u} = 0$$

where the t-vector $\underline{u} = (r-1)$.

Reference

Davis, P. J. [1963], Interpolation and Approximation, Blaisdell, Waltham.

Appendix

```
      ∇ABEC[⎕]∇
    ∇ Z←ABEC X
[1]   Z←P[1]+P[2]×*P[3]×X
    ∇


      ∇FIT1[⎕]∇
    ∇ LINE←ZA FIT1 Z;RL1;M
[1]   M←(RL1←⌹1+⍳ρZ)+.×⊖(ZA-Z)÷ZA-1
[2]   LINE←M,(⊖ZA-1)-M←M÷RL1+.×RL1
    ∇
      ∇FIT1A[⎕]∇
    ∇ NFIT1←ZA FIT1A Z;RES;N
[1]   'X-INTERCEPT IS ';-÷/φLINE
[2]   NFIT1←ZA-*((N,1)ρ⍳N←ρZ)⊥LINE
[3]   RES←(Z-NFIT1)×Z-NFIT1
[4]   'MAXSQRESIDUAL IS ';⌈/RES
[5]   'NORM RESIDUAL IS ';(+/RES)*0.5
    ∇


      ∇FIT2[⎕]∇
    ∇ P←INT FIT2 Z;YL1;R;XI
[1]   R←⍳ρYL1←Z-1
[2]   P←(-(XI+.×YL1)÷XI+.×XI),0.01 ZERO INT
[3]   P←(1-P[1]×*P[2]),P
    ∇
      ∇FIT2A[⎕]∇
    ∇ NFIT2←P FIT2A Z;RES
[1]   NFIT2←ABEC⍳ρZ
[2]   RES←(Z-NFIT2)×Z-NFIT2
[3]   'MAXSQRESIDUAL IS ';⌈/RES
[4]   'NORM RESIDUAL IS ';(+/RES)*0.5
    ∇
      ∇FN[⎕]∇
    ∇ V←FN C;ERC;ETA;ALPHA
[1]   ALPHA←(ETA←EC-R×ERC)+.×XI←EC-ERC←(EC←*C)*R
[2]   V←YL1+.×ETA-XI×ALPHA÷XI+.×XI
    ∇
      ∇ZERO[⎕]∇
    ∇ Z←TOL ZERO X;G
[1]   →ERR×⍳0<(FN X[1])×FN X[2]
[2]  BACK:→0×⍳TOL≥|G←FN Z←0.5×+/X
[3]   →BACK,X[1+(0≥G×FN X[1])]←Z
[4]  ERR:'ERROR'
    ∇


      ∇LINEFIT[⎕]∇
    ∇ LINE←ZA LINEFIT Z;T;R
[1]   Z←⊖ZA-Z
[2]   LINE←(+/R×Z)-(÷T)×(+/Z)×+/R←⍳T←ρZ
[3]   LINE←LINE÷(+/R×R)-(÷T)×(+/R)*2
[4]   LINE←LINE,(÷T)×(+/Z)-LINE×+/R
    ∇
```

May 1976
(9 June 1976)

Abduction Machines and Language Acquisition

Progress Report

by

S. Shrier

The Word Class Discovery Algorithm

A Mathematical Model

1. Consider an "incidence matrix" of word equivalents whose entries are updated as the partitioning procedure is carried out. All the entries are initially set to some number p_1, $0<p_1<1$. At each stage of the procedure, the entry corresponding to the pair of words selected for testing is either augmented or set to zero according as the test result for this pair is "believed equivalent" or "not equivalent" respectively; the other entries are unchanged. Thus the x,y entry in the matrix after the r+1st stage is

$$p_{xy}(r+1) = \begin{cases} p_{xy}(r) & \text{the pair x,y not tested} \\ 0 & \text{discovered that } x \not\equiv y \\ f(p_{xy}(r)) & \text{strengthened belief that } x \equiv y \end{cases}$$

where $f(\cdot)$ denotes a function which augments entries in the believed equivalent case.

2. The rate of convergence of this matrix to the <u>true</u> incidence matrix of the infinitary equivalence relation will be delayed by a non-zero probability of testing two words for equivalence and getting an "incorrect" result. That is, if two words x and y are not equivalent it is possible that for example, out of 100 sentences which involve the word x only 20 of these sentences would separate x from y; i.e. only 20 of these sentences would be ungrammatical with the word y substituted for the word x. This suggests that the ratio of the number of sentences which separate x from y to the total

number of sentences involving x (in the case that x and y are not equivalent) be considered as a candidate for the aforementioned probability; note that this quantity, which will be denoted by epsilon, depends on each pair of words. If epsilon is one, then all grammatical sentences involving x separate x from y when they are not equivalent; in such a case, the non-equivalent words are separated as soon as they are presented together for testing. If epsilon is zero, then every grammatical sentence involving x is also grammatical with y substituted for x and hence x and y are equivalent.

3. The rate of convergence will also depend upon the frequency with which pairs of words are brought forth for comparison with respect to the equivalence relation. For fixed words x and y which are not equivalent, it is of interest to compute the rate at which the corresponding matrix entry p_{xy} converges to zero. Since the underlying process is probabilistic, we propose to compute the mean rate at which such an entry converges to zero. This will indicate how to estimate the mean time to determine the <u>true</u> incidence matrix and hence the mean time to determine all non-equivalent words.

Let c_{xy} denote the probability that words x and y are brought forth for comparison. Then the expected value of the sum of the possible entries $p_{xy}(r)$ for non-equivalent words is estimated by

$$E\Big[\sum_{x\neq y} p_{xy}(r)\Big] = \sum_{x\neq y} E[p_{xy}(r)] \leq (\text{no.of non-eq. wds.})* \\ *\sup_{x,y}\Big\{E[p_{xy}(r)]\Big\}.$$

4. The augmentation function $f(\cdot)$ is a concave function which increases an entry in the matrix from the initial (tabula rasa) value p_1 to a value according as the strength of belief of equivalence of the corresponding pair of words increases. The r-th iterate of this function which enters the estimate of the rate of convergence behaves asymptotically like the function $1 - ab^r$, as will be shown below; such a choice is natural for the function which is to indicate increasing strength of belief in the word class equivalence of words.
Let $f^{r*}(x)$ denote the r-th iterate of the function $f(x)$ which maps the interval $[0,1]$ into itself; moreover, assume that $f(x)$ is continuous, that $f'(x)$ exists in $(0,1)$ and that $f(1)=1$. Then by the successive application of the mean value theorem,

$$\begin{aligned} f^{r+1*}(x) = f[f^{r*}(x)] &= f[f^{r*}(x)] - f(1) + f(1) \\ &= 1 - \{f(1) - f[f^{r*}(x)]\} \\ &= 1 - k_r[1 - f^{r*}(x)] \\ &= 1 - k_r\{1 - [1 - k_{r-1}(1 - f^{r-1*}(x))]\} \\ &= 1 - k_r k_{r-1}[1 - f^{r-1*}(x)] \\ &\;\;\vdots \\ &= 1 - (k_r k_{r-1} \cdots k_1)[1 - f(x)] \\ &= 1 - \Big(\prod_{i=0}^{r} k_i\Big)(1 - x) \\ &\sim 1 - ab^{r+1}, \end{aligned}$$

where the k_i are constants (values of the derivate of $f(\cdot)$ at the appropriate intermediate value).

5. Suppose that we center attention on a fixed pair of non-equivalent words x and y. After t sentences in the partitioning procedure have been processed, suppose that k of them produced the pair for comparison: then the x,y entry in the matrix

$$p_{xy}(t) = p(t) = \begin{cases} f^{k*}(p_1) \\ \text{or} \\ 0 \end{cases},$$

where $f^{k*}(\cdot)$ denotes the k-th iterate of the augmentation function. This entry is non-zero in the case that k trials took place with x and y believed equivalent; each such trial has probability $d = 1 - \epsilon_{xy}$, where epsilon was described in section 2. Hence,

$$E[p(t): \text{of which k trials involve x,y}] = f^{k*}(p_1)d^k,$$

where $0 \leq k \leq t$.

6. Recall that c denotes the probability that words x and y are brought forth for comparison; then for the t trials performed, we have

$$\begin{aligned} E[p(t)] &= (1 - c)^t + tc(1 - c)^{t-1}df(p_1) + \ldots + c^t d^t f^{t*}(p_1) \\ &= \sum_{k=0}^{t} \binom{t}{k} c^k (1 - c)^{t-k} d^k f^{k*}(p_1) \\ &\sim \sum_{k=0}^{t} \binom{t}{k} (cd)^k (1 - c)^{t-k} [1 - a\, b^k] \\ &\sim [(cd) + (1-c)]^t - a[(cdb) + (1-c)]^t \end{aligned}$$

$$\sim[(cd) + (1 - c)]^t, \quad \text{since } 0<b<1;$$

thus,

$$E[p(t)] \sim [1 - c(1-d)]^t$$

$$\sim [1 - c\epsilon]^t .$$

7. For a given pair of words which is equivalent, after t sentences the x,y entry will be

$$p_{xy}(t) = p(t) = f^{k*}(p_1)$$

for k sentences (out of the t) which involve this pair.

Then the mean value of the entry is

$$E[p(t)] = \sum_{k=0}^{t} \binom{t}{k} c^k (1 - c)^{t-k} f^{k*}(p_1)$$

$$\sim 1 - a[(cb) + (1 - c)]^t .$$

8. To summarize, after t sentences have been generated k of which involve the pair x,y, the corresponding entry in this "incidence matrix" has mean value for large number of trials t given by

$$E[p_{xy}] \sim \begin{cases} 1 - a[1 - c(1 - b)]^t & \text{if } x \equiv y \\ [1 - c\epsilon]^t & \text{otherwise.} \end{cases}$$

ABDUCTION ALGORITHMS FOR GRAMMAR DISCOVERY

by

S. Shrier

B.S., Columbia University, 1964
M.S., Columbia University, 1966

Thesis

Submitted in partial fulfillment of the requirements for the

Degree of Doctor of Philosophy in the Division of

Applied Mathematics at Brown University

June, 1977

Introduction

This paper deals with a model of language syntax acquisition. It is assumed that the artificial language has a finite description which we hope to discover on the basis of a finite sample of sentences. Specifically excluded is the simple formulation of the observations actually made, although a naive description is highly desirable. In the terminology of learning models, an insightful model is to be preferred over the rote learning exemplified in a list.

Proposal for the study of this problem was posed in the paper "Pattern Conception" by Milier and Chomsky [1957]; this paper elaborated on the virtues of finite state automaton models. An early description of a machine to carry out grammar discovery is given by Solomonoff in [1957]. Variations of this problem are found in artificial intelligence, human congnitive studies, pattern recognition, linguistics, and in systems theory under labels such as inductive inference, automaton identification and grammatical inference. The fine exposition by Fu [1974] in a chapter entitled "Grammatical Inference for Syntactic Pattern Recognition" surveys many approaches and also contains 49 references. Additional references can be found in Trakhtenbrot and Barzdin [1973]. More recent references are Adleman and Blum [1975] which deals with degrees of unsolvability of inductive inference problems and Angluin [1976] which explores complexity of the inference of finite state grammars from a finite set of positive and negative sample strings.

The following investigation is motivated toward the presentation of natural models. The investigation follows the paths initiated by Grenander [1974] in his abductory induction models. If we restrict consideration to regular grammars, the number of possible grammars is overwhelmingly large for even small size alphabets and modest numbers of variables. This implies that models for analogy to real world phenomena which exhibit language acquisition ability cannot be based on enumerative inference or finite search techniques; for this reason, we also seek an alternative to direct implementations of maximum likelihood solutions which select one grammar over another on probabilistic criteria by the solution of large scale linear systems.

In particular, algorithms are presented which carry out grammar discovery by the construction of the (right invariant) equivalence classes induced by the finite state automaton. The classes are established by means of a training sequence and a teacher. The number of possible partitions induced by an equivalence relation is known to be given by sums of Stirling numbers of the second kind. To reduce the combinatorial complexity of the problem, an equivalence relation defined on the word dictionary in a natural way by grammatical substitutability is used to partition this dictionary into grammatical equivalence classes. The prototype of each word class, that is, the representor of each of the established classes, is then used to carry out the synthesis of the minimal state automaton. The algorithms are imbedded in a statistical environment; they are studied experimentally and theoretically. Special attention is focussed on the

flexibility of the algorithms to accommodate non-systematic errors (e.g. a teacher which occasionally makes errors).

The model consists of two components: the first component is a teacher in a dual role. The teacher acts as a generative grammar which produces strings in a syntax-controlled probability language (Grenander [1967]). The random structure of the language is induced by imposing probabilities on the rules of the grammar. In this way, the production of some strings can be inhibited while the production of others can be made more likely. This adds to the teacher's grammar a facet of linguistic performance although it does not increase the generative power of the grammar. The teacher also serves as an acceptor of strings; it can judge whether or not a sentence presented to it by the learner is within its competence.

The second component, called the learner, consists of two principal procedures which carry out the construction of a copy of the teacher. The first procedure or phase is devoted to the classification of words into equivalence classes on the basis of grammatical substitutability; this procedure is in principle infinitary. These classes are familiar in structural linguistics where they are called families (Kulagina [1958]) or categories (Miller and Chomsky [1963]) although we do not use these classes in quite the same way. They are introduced here to reduce the combinatorial complexity of the constructions.

The first phase begins with the assumption that all words are in one equivalence class: this is the tabula rasa with which

the learner begins. The discovery of the classes is carried out by the resolution of the dictionary into the classes which form the required partition. The teacher randomly generates strings which are presented to the learner as a training sequence. For each string in the training sequence, a word is selected according to a weighted probabilistic strategy; this might be thought of as an attention function. This string is used to either strengthen the learner's belief about the selected word's membership in its present grammatical class or it is used to introduce a new hypothesis to be entertained about the relation of this word to the sought for partition of the dictionary. The term abduction, introduced by C.S.S. Peirce [1931] to describe the starting of a hypothesis, is applied to describe this process which either changes the class membership of the selected word or forms a new class with this selected word. All classes formed are characterized by a fixed representative word called a prototype.

The second phase carries out the discovery of the syntactic variables and the rewrite rules which govern these variables. Initially in this phase, the learner has a tabula rasa with respect to variables; the discovery of variables proceeds in a manner analogous to the phase one process. Each string in the training sequence is analyzed to some preset depth to determine initial string equivalence classes, i.e. the syntactic variables. Each string can be decoded with respect to the word class partition determined in phase one as described above. This indexing scheme is used to implement efficiently the process which determines the

partition of initial strings into the sought for syntactic variables. A subsequent encoding of initial strings is a representation of the rewrite rules. A simple tally scheme computes the experimental frequency-defined probabilities.

The model described above is a blueprint for a language discovery machine. Such a machine has been implemented in A programming Language (APL): this machine has been tested on a fragment English grammar and on several formal grammars. The fragment English grammar consists of 87 rules on 52 words in 23 classes; the rules govern the 18 syntactic variables in this syntax-controlled probability grammar. The teacher-learner interaction is protrayed with no explicit semantics and no environment. That is, (context) semantics and pragmatism are contained in neither the teacher or learner nor the training sequence. The language strings appear to have a semantic aspect: this is built into the syntactic rewrite rules. The expected sentence length (computed from the mathematical model) is 7.05 words. In a typical experiment, 115 sentences were heard by the learner to determine 20 of the 23 classes and to correctly classify 48 of the words in the dictionary. After 27 sentences were analyzed to a depth of 8 words, 17 of the 18 variables were discovered; when the depth was increased to 15, 20 additional sentences were generated before the remaining variable was discovered. The graphs in figures (1) and (2) illustrate the learning characteristics exemplified by the word class discovery procedure. More complete experimental results and a mathematical learning model appear in later sections.

Preliminaries

The following sections, which present no new results, contain the definitions and results from formal language theory and automaton theory which serve as research background for syntactic abduction of linear strings presented in the later chapters. The exposition follows Hopcroft and Ullman [1969]. The exposition of the syntax-controlled probabilities follows Grenander [1967].

Phrase Structure Languages

1. Let V_T denote a finite set of symbols called terminals or words; let V_T^+ denote the set of all finite length strings of these words; and let V_T^* denote $V_T^+ \cup \{NULL\}$ where the empty string of length zero is denoted by NULL. A language is any element of the powerset of V_T^*. Those (possibly infinite) subsets of V_T^* which have finite generating representations are called recursively enumerable. These finite generating representations or specifications are called phrase structure grammars and are formulated as follows: introduce an auxiliary finite set of symbols, denoted by V_N, called non-terminals or syntactic variables, with a distinguished symbol $S(\in V_N)$; introduce a finite set of rewrite rules R which govern these variables and which is a subset of $V_N^+ \times V^*$, where V denotes $V_N \cup V_T$. Then the phrase structure language consists of the strings which can be derived from S (the start symbol) by successive application of the rules. This is rigorously described by the introduction of a relation from V^+ to V^* as follows: for any $u \in V^+$ and $v \in V^*$, u is said to directly derive v (in the grammar) if there are strings $i,j,x,y \in V^*$ such that $u = xiy$, $v = xjy$ and $(i,j) \in R$. This can be extended by saying u derives v in the grammar if either u=v or if there is a finite sequence $Z_0, Z_1, Z_2, \ldots, Z_m \subset V^*$, $m > 1$, such that $u=Z_0$, $v=Z_m$, and Z_i directly derives Z_{i+1} for $i=0(1)m-1$. Then the language generated by this grammar is defined as the set of strings of words which can be derived from the start symbol S.

If S derives u (denoted by $S \rightarrow u$) and u contains variables, then u is called a sentential form. Note that the elements in the set of rewrite rules, for example denoted by (i,j), are customarily also written as $i \rightarrow y$. If two grammars generate the same language, then they are said to be weakly equivalent.

2. Context-Free Languages. If the rewrite rules R are restricted to finite subsets of $V_N \times V^*$, then the resultant grammar is called a context-free grammar.

3. Finite-State Languages. Those subsets of context-free grammars to which we focus our attention are called finite state grammars. The variables are governed by rewrite rules of two types: continuing $u \rightarrow xj$ or terminating $i \rightarrow x$, where $i,j \in V_N$ and $x \in V_T$. Denote the finite set of rules which rewrite i by R_i and assume that the generating algorithm begins by the application of a rule selected from R_1.

The language generated by such a grammar, which consists of V_T, V_N and R, is denoted by L(G); the symbol G denotes the triple (V_T, V_N, R) and R denotes the finite set of all rules. The language L(G) consists of all those strings in V_T^* which can be produced by the application of the rules in R; note that any string in L(G) is as likely to appear as any other as a result of the application of the grammar G.

4. Relation to Finite State Automata. The language generated by a grammar is some set of strings as described above. This set is also the set accepted by some finite state automaton.

The identification of L(G) with its finite state automaton acceptor proceeds as follows:
the variables in V_N constitute the states $1,2,\ldots,n_v$; in addition, introduce a final state F which is the "target" state for those variables which are governed by terminating rewrite rules.

5. Syntax-Controlled Probabilities. For each $R_i = \{r_{i_1}, r_{i_2}, \ldots, r_{in_i}\}$, where $r_{..}$ denotes a rule and n_i is the number of rules rewriting i introduce a probability distribution over R_i so that

$$\sum_{j=1}^{n_i} P(r_{ij}) = 1,$$

where $i=1,2,\ldots,n_v$.

6. Markov Chain. Consider the application of the grammar G together with the probability distribution. In terms of the automaton description, the probability that the machine will be at state ℓ at time t+1 given that it is at state k at time t is specified. A system which evolves through a finite number of states (n_v+1) with a specified conditional probability of transition between two states for a given state at time t which is independent of t is called a finite homogeneous Markov chain (Kemeny and Snell [1960]). The familiar state diagram for finite automata (with labeled arcs which indicate letters in V_m and the probability) has a description in terms of two matrices: a matrix of probabilities and a matrix which prescribes the letters which correspond to transitions between states.

ADELMAN, L. AND M. BLUM (1975) "INDUCTIVE INFERENCE AND UNSOLVABILITY", DEPT. OF E. E. UNIVERSITY OF CALIFORNIA BERKELEY (INTERNAL DOCUMENT)

ANGLUIN, DANA (1976) "AN APPLICATION OF THE THEORY OF COMPUTATIONAL COMPLEXITY TO THE STUDY OF INDUCTIVE INFERENCE", MEMORANDUM ERL-M586, UNIVERSITY OF CALIFORNIA (BERKELEY) COLLEGE OF ENGINEERING

DAVIS, P. J. (1963) INTERPOLATION AND APPROXIMATION, BLAISDELL, WALTHAM

FU, K. S. (1974) SYNTACTIC METHODS IN PATTERN RECOGNITION, ACADEMIC PRESS, NEW YORK

GRENANDER, U (1967) "SYNTAX-CONTROLLED PROBABILITIES", REPORTS ON PATTERN ANALYSIS #2, DIVISION OF APPLIED MATHEMATICS, PROVIDENCE

---------- (1974) "ABDUCTION MACHINES THAT LEARN SYNTACTIC PATTERNS", REPORTS ON PATTERN ANALYSIS # 25, DIVISION OF APPLIED MATHEMATICS, PROVIDENCE

HOPCROFT, JOHN E. AND JEFFREY D. ULLMAN (1969) FORMAL LANGUAGES AND THEIR RELATION TO AUTOMATA, ADDSSON-WESLEY, READING

JOHNSTON, JOHN B. (1971) "THE CONTOUR MODEL OF BLOCK STRUCTURED PROCESSES", PROC. SYMP. ON DATA STRUCTURES IN PROGRAMMING LANGUAGES, SIGPLAN NOTICES, ACM, NEW YORK, 55-82

KEMENY, JOHN AND LAURIE J. SNELL (1960) FINITE MARKOV CHAINS, VAN NOSTRAND, NEW YORK

KULAGINA, O. S. (1958) "OB ODNOM SPOSOBE OPREDELENIJA GRAMMATIČESKIH PONJATIĬ NA BAZE TEORIĬ MNOZESTV" PROBLEMY KIBERNETIKI, T. 1, 203-214

MILLER, G. A. AND CHOMSKY (1957) "PATTERN CONCEPTION" ASTIA DOCUMENT AD110 07 AIR FORCE CAMBRIDGE RESEARCH CENTER, BEDFORD

---------- (1963) "FINITARY MODELS OF LANGUAGE USERS", HANDBOOK OF MATHEMATICAL PSYCHOLOGY V. 2 (ED LUCE, BUSHM GALANTER) WILEY, NEW YORK 419-491

SOLOMONOFF, R. J. (1957) "AN INDUCTIVE INFERENCE MACHINE", IRE NATIONAL CONVENTION RECORD V, PART 2 SESSION 22

TRAKHTENBROT, B. A. & JA. M. BARZDIN (1973) FINITE AUTOMATA: BEHAVIORAL